



**UNIVERSITY  
OF ICELAND**

# Greining Reiknirita Lokaverkefni

Kristján Sölvi Örnólfsson  
Skjöldur Orri Eyjólfsson

13. apríl 2025

# Efnisyfirlit

<b>Inngangur</b>	<b>1</b>
<b>Aðferðir</b>	<b>1</b>
Gagnameðhöndlun . . . . .	1
Netið . . . . .	2
Reiknirit . . . . .	2
<b>Niðurstöður</b>	<b>3</b>
Uppbygging netsins . . . . .	3
Árangur reiknrita . . . . .	3
Tímamælingar . . . . .	4
<b>Samantekt og næstu skref</b>	<b>4</b>
Samantekt á niðurstöðum . . . . .	4
Takmarkanir á aðferðafræði/lausnaraðferðum . . . . .	4
Næstu Skref . . . . .	5
<b>Heimildir</b>	<b>5</b>
<b>Samstarf</b>	<b>5</b>

## Inngangur

Eftirfarandi verkefni var hugsað til þess að við myndum læra að nota aðferðir sem við höfum lært í námskeiðinu í raunverulegum verkefnum. Verkefnið sem við völdum fólst í því að gera leitarvél fyrir strætóferðir með því að nota gögnin sem Strætó birtir opinberlega.

Hugmyndin var að meðhöndla og skipuleggja gögnin á þann hátt að hægt væri að smíða nákvæmt net, sem innihéldi bæði hefðbundnar strætóleiðir og möguleika á gönguleiðum á milli nálægra stoppi-stöðva. Netið var svo notað til að gera aðlagðar útgáfur af þekktum reikniritum, eins og Dijkstra og A\*, til að leysa mismunandi tilvik ferðaparfa, frá einföldum beinum leiðum til flóknari leiða sem kröfðust færri skiptinga eða nýttu gönguleiðir til að skera niður heildarferðatíma.

## Aðferðir

### Gagnameðhöndlun

Í þessu verkefni átti að gera leitarvél fyrir Strætó, til þess var notaðst við raungögn frá heimasíðu strætó. Gögnunum var skipt niður í töflur eftir því hverju þau voru að lýsa. Til einföldunar voru mikilvægustu töflurnar tengdar saman (þ. e. routes, trips og stop times) til að hægt væri að vinna með eina töflu í stað þess að vinna með margar á sama tíma. Því næst voru ákveðna raðir fjarlægðar, t.d. næturstætó og strætó leiðir sem fóru ekki á virkum dögum, þetta var gert til einföldunar. Að lokum tengdum við tvær eins töflur (köllum þær tafla 1 og tafla 2) saman á eftirfarandi hátt: Röð í

töflu 1 var tengd við röð í töflu 2 ef `trip_id` var eins í báðum töflum og ef `stop_sequence` í töflu 2 var einum hærra heldur en í töflu 1. Þá fékkst lokatafla þar sem sérhver röð táknaði ferð á milli tveggja stoppistöðva.

## Netið

Við notuðum þessa fyrrnefndu lokatöflu til þess að búa til netið. Netið  $G$  var skilgreint á eftirfarandi hátt. Hver einasta stoppustöð var hnútur og ef vagn fór frá stoppustöð  $u$  á tíma  $t_1$  og að stoppustöð  $v$  á tíma  $t_2$  með engum öðrum stoppustöðvum þar á milli að þá var búinn til nýr millihnútur,  $w$  ásamt nýjum leggjum  $(u, w)$  og  $(w, v)$  með tímum  $t_1$  og  $t_2$  sem vigtir. Við notuðum uppflettistöflu (e. dictionary) í Python til þess að setja fram netið, þ.a. hver lykill í uppflettistöflunni var einhver hnútur  $v$  í netinu. Þessir lykilar höfðu síðan gildi sem var önnur uppflettitafla þar sem lykarnir í henni voru þeir hnútar sem höfðu legg til sín frá þessum hnúti  $v$ . Þessir lykilar í innri uppflettistöflunni höfðu síðan gildi sem svaraði til vigtar leggjarins. Lyklarnir fyrir hnútana sem táknuðu stoppistöðvar voru einfaldlega `stop_id`. Lyklarnir fyrir hnútana sem táknuðu strætó ferðir voru `trip_id` og `stop_sequence` með niðurstriki á milli.

Seinna í verkefninu átti leyfa þann möguleika að labba á milli stoppistöðva ef þær voru nálægt hvor annarri, en við völdum 500 metra. Til þess að þetta væri hægt þurfti að breyta netinu. Við bættum við "gönguleggjum á milli stoppistöðva ef fjarlægðin á milli þeirra var minni en 500 metrar. Þyngd "gönguleggjanna voru þá tíminn í mínútum sem það tæki að labba að stoppustöðinni á  $5\text{km/klst}$ . Til þess að gera greinarmun á þessum leggjum bættum við merkingu sem sagði til um hvort leggurinn táknaði labb eða strætóferð. Notast var við formúlu Haversine sem reiknar stystu fjarlægð milli tveggja punkta á hnetti, en þar sem þetta voru mjög stuttar fjarlægðir hefði alveg fullnægjandi að nota einfalda nálgun til að spara keyrslutíma, en það skipt engu máli fyrir leitartímann þar sem að þetta er gert einungis þegar netið er búið til.

## Reiknirit

Nokkur tilfelli voru fengist við í þessu verkefni. Einfaldasta tilfellið var að finna stystu leið frá stoppustöð  $A$  og að stoppustöð  $B$  með upphafstíma  $t$  án þess að taka tillit til þess að hægt sé að labba á milli stoppistöðva. Til að leysa þetta verkefni var notað net eins og var lýst að ofan án gönguleggjana. Reikniritið sem var notað líktist Dijkstra en þó með þeirri breytingu að í stað þess að slaka á leggjum með að leggja saman þyngdir þá notum við löglega komutíma. Til að getið lesið komutíma og brottferðatíma þá bjuggum við til hjálparfall sem breytti þessum tímum í fjölda mínútna eftir miðnætti.

Næst var skoðað hvernig hægt væri að leysa verkefnið með það í huga að mögulegt væri að ganga á milli stoppistöðva sem eru í hæfilegri göngu fjarlægð. Til að leysa þetta verkefni var notað mjög svipað reiknirit og áður nema með þeirri breytingu að netið sem var notað innihélt gönguleggi. Meginmunurinn á göngu legg og strætó legg er sá að það er alltaf hægt að fara göngu legg en ekki alltaf hægt að fara strætó legg, (stundum er nauðsynlegt að bíða eftir strætó). Því er hægt að skoða göngu leggi beint þ.e án þess að bíða eftir þeim.

Stundum eru margar ólíkar leiðir sem skila sömu niðurstöðu og því var hugmyndin að gera reiknirit

sem skilar þá leiðinni sem inniheldur fæstar skiptingar. Þetta var gert með því að halda utan um fjölda ólíkra strætó ferða fyrir hverja leið í Dijkstra. Svo fyrir tvær mögulegar leiðir þá var borið saman fjölda ólíkra strætó leiða og síðan uppfært leiðina ef fjöldi ólíkra strætó leiða fyrir nýju leiðina var minni.

Heldur minna þekkt reiknirit en Dijkstra en þó vel þekkt er reikniritið sem er gjarnan kallað A\*[1]. Hugmyndin á bakvið það er að í staðinn fyrir að skoða alla hnúta eins og í Dijkstra þá skoðum við hnútana í þeirra röð sem röð "skynsamleg". Það er gert með því að búa til svokallað fjarlægðarfall sem metur fjarlægð frá hnút í lokahnútinn. Það sem nauðsynilegt er að hafa í huga þegar fallið er valið er að passa að það gefi niðurstöðu sem er alltaf betri en raunverulega niðurstaðan frá hnút yfir í lokahnút og að fallið sé minnkandi. Í þessu verkefni var notað fall sem skilaði niðurstöðu af beinni línu frá stoppustöð í aðra og gert ráð fyrir 90km/klst keyrslu þá leið (hámarkshraði til þess að uppfylla skilyrðið  $w(u, v) + f(v) \geq f(u)$  ( $w(u, v)$  er vigt leggjarins frá  $u$  til  $v$  og  $f$  er fjarlægðarfallið) vegna þess hvorki strætó né gangandi vegfarandi fer hraðar en 90 km/klst að meðaltali og því er skilyrðið uppfyllt).

Framsetningin á niðurstöðu leitarinnar var nokkuð einföld. Þegar reikniritin höfðu skilað stystu leið, sem var þá bara röð hnúta í netinu frá upphafs- til endahnúts, að þá var farið í gegnum þessa hnúta og var stop\_id notað til að fletta upp nafninu á stoppistöðinni og trip\_id notað til að fletta upp strætóleiðinni. Ef trip\_id breyttist að þá þýddi það að það þurfti að skipta um strætóleið og ef tveir hnútar sem táknuðu stoppistöð voru í röð að þá þýddi það að það þurfti að labba á milli þeirra.

Til þess að fá út sem besta tímamælingu voru valdar 50 mismunandi samsetningar á brottfararstöð og komustöð af handahófi. Fyrir hverja samsetningu voru bætt við 5 ólíkir brottfarartímar, þ.e.  $50 \cdot 5 = 250$  ólík inntök fyrir leitarreikniritin.

## Niðurstöður

### Uppbygging netsins

Netið okkar innihélt 60615 hnúta og stoppustöðin með hæstu útgráðuna, þ.e fjöldi ferða sem fara frá stoppustöðinni, var Mjódd með 559 útleggi.

### Árangur reiknirita

Leitarvélin okkar nýtti aðlagðar útgáfur af Dijkstra og A\* reikniritum til að finna bestu mögulegu leiðirnar í samgöngukerfinu. Niðurstöður sýndu að reikniritin skiluðu leiðum sem voru réttar og oft á tíðum svipaðar þeim sem leitin á heimasíðu Strætó bauð upp á.

Leiðin frá Meistaravöllum til FB var skoðuð fyrir öll mismunandi reikniritin. Einnig voru aðrar leiðir skoðaðar til þess að ganga úr skugga um að reikniritið væri rétt. Við notkun hefðbundins Dijkstra reiknirits tók leiðin frá Meistaravöllum til FB lengri tíma, komutími kl. 12:31, og krafðist skiptinga sem voru tímafrekar. Með því að innleiða möguleika á gönguleiðum milli stoppistaða sem eru nálægt hvor öðrum tókst okkur að stytta ferðatímann marktækt, komutími var kl. 12:23. Að auki, þegar reikniritið var stillt á að lágmarka fjölda skiptinga, fundum við leiðir sem þóttu notendavænni

þar sem þær krefjast færri skiptinga á strætóvögnum, sem leiðir til einfaldari og þægilegri ferðalaga fyrir farþega.

Við útfærslu á A\* reikniritinu kom í ljós að það tók tvöfalt lengri tíma en venjulegt Dijkstra reiknirit, þrátt fyrir að bæði reiknirit skiluðu sömu leiðinni. Þessi tímamunur kom okkur á óvart þar sem A\* er almennt þekkt fyrir að vera hraðvirkara þar sem reikniritið "þvingar" til þess að skoða frekar þær leiðir sem eru sífellt að nálgast endahnútinn. Við frekari greiningu grunuðum við að fjarlægðarfallið, sem er mikilvægt fyrir ákvörðun A\* reikniritsins, hafi ekki verið nákvæmt eða viðeigandi fyrir þetta sérstaka notkunarvið. Þegar fjarlægðarfallið var sett sem 0, sem leiðir til þess að A\* verður eins og Dijkstra, að þá skiluðu tímamælingarnar samt svipaðri niðurstöðu sem bentu til þess að það var eitthvað annað en fjarlægðarfallið sem var að valda því að reikniritið var ekki jafnhratt og við mátti búast. Þrátt fyrir að hafa lúslesið kóðann oft að þá tókst okkur aldrei að finna út úr því hvers vegna A\* virkaði ekki sem skyldi.

## Tímamælingar

Tímamælingar fyrir útfærsluna sem innihélt gönguleiðir sýndu að hægt var að meðhöndla um þrjár fyrirspurnir á sekúndu. Sem við teljum að myndi vera nægilega gott fyrir flesta sem væru að leita að strætó leið. Við reyndum eftir okkar bestu getu að auka afköst reikniritanna. Dæmi um það var að í staðinn fyrir að klára að keyra Dijkstra fyrir alla hnútana að þá skiluðum við úr leitarfallinu um leið og við fengum endahnútinn úr forgangsbiðröðinni, enda engin ástæða þá að halda áfram að skoða fleiri hnúta. Við notuðum einnig uppflettitöflur mikið í staðinn fyrir að sækja úr pandas dataframe, t.d. þegar það þurfti að fletta upp stop\_id fyrir stop\_name.

## Samantekt og næstu skref

### Samantekt á niðurstöðum

Við notuðum aðlagðar útgáfur af Dijkstra og A\* reikniritum til að ákvarða hentugustu leiðirnar í Strætó kerfinu. Reikniritin skiluðu réttum leiðum, oft líkar þeim sem finnast í leit á heimasíðu Strætó. Sérstaklega var skoðuð leiðin frá Meistaravöllum til FB:

- **Án gönguleiða** Tók lengri tíma með kostnaðarsömum skiptingum.
- **Gönguleiðir leyfðar** Innleiðing gönguleiða stytta tíman verulega
- **Færri skiptingar** Reikniritið sem lágmarkaði strætó skipti fann einfaldari og þægilegri leiðir.

Þegar A\* var notað tók það óvænt tvöfalt lengri tíma en Dijkstra, þrátt fyrir að skila sömu leiðum.

### Takmarkanir á aðferðafræði/lausnaraðferðum

Megin takmarkanirnar á lausnaraðferðunum eru eftirfarandi

- 1) **Strætóleiðir.** Í verkefninu þá var til einföldunar tekið út alla næturstrætóa og helgarstrætóa. Það að hafa alla strætóa myndi leiða til flóknara nets og reiknirits þar sem að það þyrfti að

halda utan um hvaða vikudagur væri.

- 2) **Leit í tímatöflu.** Þegar við leitum að strætóleiðum þá notum við tímatöflu sem gerir ekki ráð fyrir að leiðir geti verið felldar niður eða seinkaðar.

## Næstu Skref

Ef tækifæri fengist væri gaman að byggja ofan á þetta verkefni. Það væri t.d. hægt að bæta leitina enn frekar með því að nota rauntíma gögn eins og GPS gögn sem segja til um hvar strætisvagnar séu staddir og/eða gögn frá Google Maps sem segja til u.þ.b. hversu mikil umferðin er á ákveðnum stöðum. Þetta myndi vera flókið verkefni en myndi hafa meira notagildi í raunveruleikanum. Síðan væri hægt forrita notendaviðmót þótt það tengist ekki þessu námskeiði.

## Heimildir

- [1] Wikipedia contributors. A\* search algorithm. [https://en.wikipedia.org/w/index.php?title=A\\*\\_search\\_algorithm&oldid=1217507422](https://en.wikipedia.org/w/index.php?title=A*_search_algorithm&oldid=1217507422), 2024. Accessed: 6 April 2024.

## Samstarf

Áhersla var lögð á jafna skiptingu á verkefnum á milli okkar þar sem við tókum hvor um sig þátt í öllum meginþáttum verkefnisins. Þó má segja að Skjöldur hafi gert meira í verkefnunum færri skiptingar (\*\*) og framsetning (\*) en Kristján í A\* (\* \* \*).